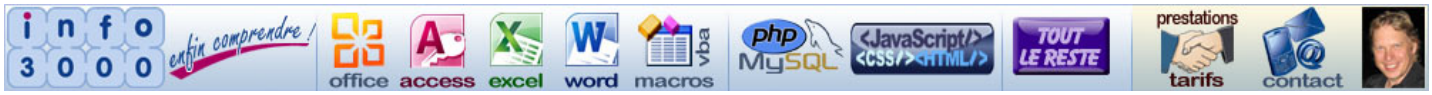


PHP - MySQL [Retour au menu]
Protection de pages HTML

S'il est très intéressant de mettre à disposition des pages HTML à disposition de tout le monde sur internet, il peut être tout aussi intéressant de limiter leur visualisation aux seuls internautes qui se sont dûment inscrits (Nom d'utilisateur, mot de passe et E-Mail)

Le cas d'école suivant vous est proposé : Vous avez composé une page HTML qui contient des informations techniques que vous ne voulez pas divulguer à d'autres personnes que celles qui se sont enregistrées : Pour l'exemple, je vais vous parler d'un poisson tout droit sorti de mon imagination : le Patrouchon. J'ai donc fait des études très approfondies :-) sur cette étrange bestiole, et je ne veux pas que n'importe qui puisse avoir accès à la fiche descriptive !!!

Explications générales

A la base, nous serons dans un simple formulaire `menu.htm` qui va donner 2 choix :

1. Entrée de son Login et mot de passe (si on est déjà inscrit)
2. S'inscrire (si on est nouveau)

Si on choisit de se logger, au moment ou on clique sur OK, le Login et le mot de passe sont transmis à la page qui contient la fichier du patrouchon `fichepatrouchon.php`. C'est dans cette page qu'une parocédure PHP va déterminer su l'utilisateur à entré un Login et un mot de passe valide (il va donc y avoir une vérification dans la base de données `info3000`, dans la table `T_Patrouchon`). Si le Login et le mot de passe sont valides, la fiche du patroucon apparait en clair, avec en bas, un lien qui retourne vers le menu (pas très important). Par contre, si le Login et/ou le mot de passe sont invalides, on n'affiche pas du tout les données techniques du patrouchon, mais un simple message "Votre login/mot de passe sont invalides", et un lien vers le retour au menu. On va évidemment utiliser un if, et un accès dans la base de données avec SQL.

Comme PHP est un langage qui s'exécute au niveau serveur, et pas directement sur le navigateur, si le Login/mot de passe ne sont pas valides, non seulement la fiche technique sur le patrouchon n'est pas visualisée, mais elle n'est pas du tout visualisable, même en allant dans le menu du navigateur Affichage/Source : Impossible de tricher !

Revenons au menu : Si l'utilisateur est un nouvel utilisateur, il va cliquer sur "Nouveau venu ?", ce qui va l'envoyer dans un formulaire d'inscription `inscription.htm`, ou il va être invité a entrer son nom (qui servira de Login), son mot de passe et son E-Mail (l'E-Mail ne servira pas dans la validation par la suite, mais c'est toujours sympathique d'avoir une manière de contacter ses visiteurs). En cliquant sur OK, nous allons vérifier que son login n'existe pas déjà dans la base, parce que sinon, ça risque de mal se passer ! C'est `verification.php` qui va soit :

- Accepter le nouveau venu, et proposer un lien vers le menu principal `menu.htm` (d'où il pourra se loguer avec le profil qu'il vien de se créer)
- Refuser pour cause de déjà existant, et proposer un lien qui retourne vers le formulaire d'inscription `inscription.htm`

Inventaire des choses nécessaires

Dans notre projet, il y aura donc :

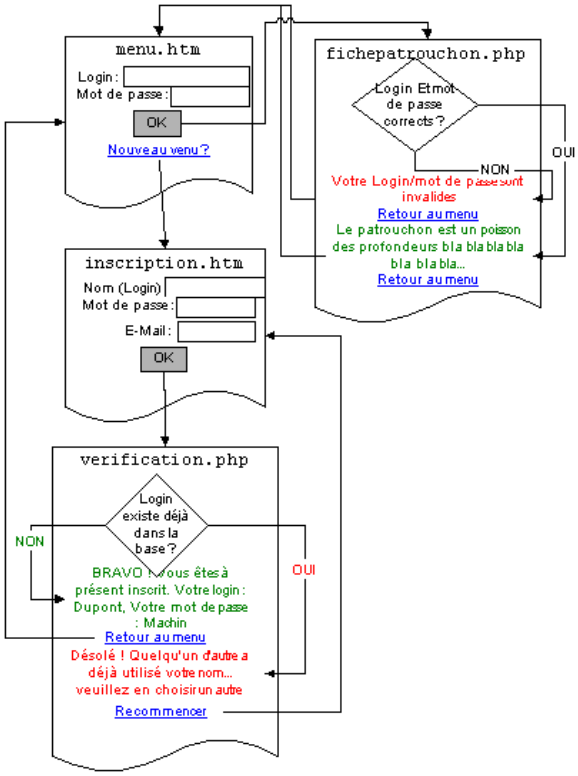
- Une base de données (`info3000`)
- Une table **`T_Patrouchon`**, Dont voici la création en SQL :

```
CREATE TABLE `ZZ_Patrouchon`
  (
  `NomVisiteur` VARCHAR(30) NOT NULL,
  `MotDePasse` VARCHAR(10) NOT NULL,
  `EMail` VARCHAR(30) NOT NULL,
  PRIMARY KEY (`NomVisiteur`)
  );
```

et 4 pages :

1. **`menu.htm`** qui va proposer de se loguer ou de s'inscrire
2. **`inscription.htm`** qui va proposer un formulaire d'inscription
3. **`verification.php`** qui va soit refuser l'inscription s'il s'agit d'un doublon, soit l'accepter et l'ajouter dans T_Patrouchon
4. **`fichepatrouchon.php`** qui va soit afficher la fichier technique du patrouchon, soit refuser tout affichage pour cause de Login incorrect

Schéma



Partie 1 : Conception du menu principal `menu.htm`

Rien de spécial à signaler : Un bête formulaire pourvu de 2 champs : EDILogin et EDIMotDePasse, ainsi qu'un lien pour les nouveaux venus

```
<body>
  <h1>Menu principal</h1>
  <form method="post" action="fichepatrouchon.php">
    Login :          <input type="text"   name="EDILogin">     <br>
    Mot de passe :   <input type="text"   name="EDIMotDePasse"><br>
                     <input type="submit" value="OK">
  </form>

  <a href="inscription.htm">Nouveau venu ?</a>
</body>
```

Vous pouvez cliquer ici pour vous rendre réellement sur ce formulaire.

Partie 2 : Conception du formulaire d'inscription `inscription.htm`

Pour les nouveaux venus, il va être proposé d'entrer leur Nom (Login), Mot de passe et E-Mail. Pour la simplicité de l'exemple, il ne sera pas contrôlé si un ou plusieurs champs sont laissés vides

```
<body>
  <h1>Veuillez remplir les champs suivants</h1>
  <form method="post" action="verification.php">
    Votre nom (qui servira de login) : <input type="text" name="EDILogin">     <br>
    Mot de passe                     : <input type="text" name="EDIMotDePasse"><br>
    E-Mail                           : <input type="text" name="EDIEMail">     <br>
                                       <input type="submit" value="OK">
  </form>
</body>
```

Partie 3 : Conception de la page de vérification de la nouvelle inscription `verification.php`

Si nous sommes sur cette page, c'est que nous provenons de `inscription.htm`, et donc nous sommes en possession des 3 variables qui proviennent des champs respectifs `EDILogin`, `EDIMotDePasse` et `EDIEMail`. Nous n'allons pas contrôler s'ils sont laissés vides pour une question de simplicité. Nous allons nous contenter de vérifier si par hasard le Login demandé n'existe pas encore dans la table `ZZ_Patrouchon`.

Il va falloir une ligne SQL qui contient dynamiquement le nom demandé (`EDILogin`) , de cette manière :

```
SELECT * FROM `ZZ_Patrouchon` WHERE `NomVisiteur` LIKE 'Michel'
```

Le problème, c'est que ce n'est pas Michel, on écrit donc :

```
SELECT * FROM `ZZ_Patrouchon` WHERE `NomVisiteur` LIKE '$EDILogin'
```

De là, c'est très simple de récupérer le nombre d'enregistrements correspondant à cette requête : S'il y en a 1, c'est que le nom est déjà pris, tout simplement ! Donc :

```
$Resultat = mysql_query("SELECT * FROM `ZZ_Patrouchon` WHERE `NomVisiteur` LIKE
'$EDILogin'");
```

Contient un pointeur sur ce résultat, et l'instruction :

```
mysql_numrows($Resultat)
```

Contient le nombre d'enregistrements résultants (en fait 0 ou 1...)

Ce qui nous permet d'écrire :

```
$Resultat = mysql_query("SELECT * FROM `ZZ_Patrouchon` WHERE `NomVisiteur` LIKE
'$EDILogin'");
if (mysql_numrows($Resultat) == 0)
  echo "OK. Ce nom est libre";
else
  echo "Désolé... ce nom est déjà pris";
```

En fait, ce ne sont pas 2 simples petites phrases que nous voulons afficher, mais du texte un peu mieux mis en forme avec HTML, suivis d'un lien hypertexte qui poiinte soit vers le menu en cas de succès, soit vers le formulaire d'inscription si le nom existait déjà. :

Bravo ! Vous êtes à présent inscrit ! Votre login est michel. N'oubliez pas votre mot de passe 3mBtt63rE !
Cliquez ici pour retourner au menu

ou

Désolé ! Quelqu'un d'autre a déjà utilisé ce nom. Veuillez en choisir un autre s'il vous plait

ce qui donne quelque chose comme ceci :

```
if (mysql_numrows($Resultat) == 0)
  echo "<h2><font color='#009900'>Bravo ! Vous êtes à présent inscrit ! Votre login est
$EDILogin. N'oubliez pas votre mot de passe $EDIMotDePasse !</font><br> <a
href='menu.htm'>Cliquez ici pour retourner au menu</a></h2>";
```

Nous avons vu en étudiant la fonction if qu'il était possible d'intégrer directement du HTML en arrêtant la balise ?> juste avant de le composer, et de le reprendre ensuite. J'ai préféré utiliser la fonction echo, simplement parce que j'utilise dans cette phrase les 2 variables $EDILogin et $EDIMotDePasse. Par contre, je vais extraire le code HTML pour le else, histoire de changer un peu :

```
if (mysql_numrows($Resultat) == 0)
  echo "<h2><font color='#009900'>Bravo ! Vous êtes à présent inscrit ! Votre login est
$EDILogin. N'oubliez pas votre mot de passe $EDIMotDePasse !</font><br> <a
href='menu.htm'>Cliquez ici pour retourner au menu</a></h2>";
else
  {// fin de la balise ouvrante qui se trouve bien avant ce code d'exemple :
  ?>
    <h2><font color="#FF0000">Désolé ! Quelqu'un d'autre a déjà
utilisé ce nom. <a href="inscription.htm">Veuillez en choisir un autre</a>
s'il vous plait</font></h2>
  <?
  }
  ?>
```

Maintenant, il ne suffit pas de dire que c'estv tout bon, encore faut-il ajouter le nouvel arrivant dans la table ZZ_Patrouchon. Nous allons utiliser une commande INSERT sur ce modèle (précisoons l'ordre des champs `NomVisiteur`, `MotDePasse`, `EMail` pour plus de clarté) ::

```
mysql_query("INSERT INTO `ZZ_Patrouchon` (`NomVisiteur`, `MotDePasse`, `EMail`) VALUES
('Patrick', 'kci', 'patrick@mail.com');");
```

Bien entendu, nous allons remplacer les valeurs Patrick, kci et patrick@mail.com par les variables respectives :

```
mysql_query("INSERT INTO `ZZ_Patrouchon` (`NomVisiteur`, `MotDePasse`, `EMail`) VALUES
('EDILogin', 'EDIMotDePasse', 'EDIEMail');");
```

Qui est dans le if :

```
if (mysql_numrows($Resultat) == 0)
  {
  mysql_query("INSERT INTO `ZZ_Patrouchon` (`NomVisiteur`, `MotDePasse`, `EMail`) VALUES
('$EDILogin', '$EDIMotDePasse', '$EDIEMail');");
  echo "<h2><font color='#009900'>Bravo ! Vous êtes à présent inscrit ! Votre login est
$EDILogin. N'oubliez pas votre mot de passe $EDIMotDePasse !</font><br> <a
href='menu.htm'>Cliquez ici pour retourner au menu</a></h2>";
}
```

Et voilà ! C'est terminé. Voici le code complet de la page `verification.php` : (les lignes en bleu sont du simple code HTML)

```
<body>
  <?
    // Initialisation habituelle :
    mysql_connect("localhost", "info3000", "MonMotDePasse");
```

```
    mysql_select_db("info3000");
    // Création de la requête pour savoir s'il y a déjà quelqu'un du même nom dans la table
:
    $Resultat = mysql_query("SELECT * FROM `ZZ_Patrouchon` WHERE `NomVisiteur` LIKE
'$EDILogin'");
    // Si le résultat est 0, tout baigne dans l'huile :
    if (mysql_numrows($Resultat) == 0)
      {
      mysql_query("INSERT INTO `ZZ_Patrouchon` (`NomVisiteur`, `MotDePasse`, `EMail`)
VALUES ('$EDILogin', '$EDIMotDePasse', '$EDIEMail');");
      echo "<h2><font color='#009900'>Bravo ! Vous êtes à présent inscrit ! Votre login est
$EDILogin. N'oubliez pas votre mot de passe $EDIMotDePasse !</font><br> <a
href='menu.htm'>Cliquez ici pour retourner au menu</a></h2>";
      }
    // Sinon, c'est qu'il y a déjà quelqu'un, il faut arrêter le pauvre nouveau venu :
    else
      {
  ?>
      <h2>
        <font color="#FF0000">
          Désolé ! Quelqu'un d'autre a déjà utilisé ce nom.
          <a href="inscription.htm">
            Veuillez en choisir un autre
          </a>
          s'il vous plait
        </font>
      </h2>
  <?
      }
    // Il est tard, il est temps de fermer :
    mysql_close();
  ?>
</body>
```

Partie 4 : Conception de la page `fichepatrouchon.php`

Dès le départ de cette page, il s'agit de voir s'il existe bien un certain "Robert" (Si au menu de démarrage, c'est Robert qui a été entré), et ensuite, si Robert existe bien, est-ce que le mot de passe fourni est bien le mot de passe correspondant à Robert ? En un mot comme en 100, quel ce soit le Login ou le mot de passe qui sont incorrects, retour au menu ! SI tout est OK, affichage de la fiche technique et retour au menu.

Nous allons utiliser la même technique que dans `verification.php` : On va compter le nombre d'enregistrements dont le nom = Robert (en fait dont le nom = `EDILogin` de `menu.htm`), mais cette fois, le mot de passe doit également exister ! Ca va se résumer comme ceci :

```
$Resultat=mysql_query("SELECT * FROM `ZZ_Patrouchon` WHERE `NomVisiteur` LIKE '$EDILogin'
AND `MotDePasse` LIKE '$EDIMotDePasse'");
if (mysql_numrows($Resultat) == 1)
  {
  echo "OK vous pouvez voir";
  }
else
  {
  echo "Non, vous n'êtes pas inscrit";
  }
```

Tout était dans la requête SQL qui vérifie tout en même temps : le login et le mot de passe.

Maintenant, il suffit de remplacer `OK vous pouvez voir` par tout le texte et les mises en forme de la fiche du patrouchon et `Non, vous n'êtes pas inscrit` par un renvoi vers le menu, avec également du HTML. Voici le code complet de la page (En bleu le code HTML pur et dur):

```
<html>
  <head>
    <title>Les Patrouchons</title>
  </head>
  <body>
    <?
    mysql_connect("localhost", "info3000", "MonMotDePasse");
    mysql_select_db("info3000");
    $Resultat=mysql_query("SELECT * FROM `ZZ_Patrouchon` WHERE `NomVisiteur` LIKE
'$EDILogin' AND `MotDePasse` LIKE '$EDIMotDePasse'");
    if (mysql_numrows($Resultat) ==1)
      {
    ?>
<h1 align="center"><font color="#0000FF">Les Patrouchons</font></h1>
<h2><i><font color="#666666">Les Patrouchons sont de petits animaux ambidextres
d'une vingtaines de centim&egrave;tres environ.</font></i></h2>
<p>Ils se d&eacute;placent &agrave; l'aide de 4 nageoires dorsales dans le fond
de la fosse des Mariannes. Ils se nourissent de petits crustac&eacute;s et de
```

```
sable.</p>
<p>Sa p&eacute;riode de reproduction se situe entre <b><font color="#FF0000">d&eacute;but
d&eacute;cembre et fin janvier</font></b>. Il s'accouple avec des patrouchonnes
au hasard de ses p&eacute;rigrinations, mais une fois qu'il en a f&eacute;cond&eacute;e
une, il s'en tient l&agrave;, et aide sa femelle &agrave; &eacute;lever ses
petits.</p>
<p>Elle pond des oeufs (4 &agrave; 6 en moyenne) au bout de 80 jours.</p>
<p>Patrouchon m&acirc;le de 4 mois environ : <img src="patrouchonphoto.gif" width="264"
height="164" align="absmiddle"></p>
<p><a href="menu.htm">Retour au menu</a></p>
    <?
      }
    else
      {
    ?>
<p><b><font color="#FF0000">Login et/ou mot de passe incorrect. <a href="menu.htm">Veuillez
recommencer votre identification SVP</a> </font></b></p>
    <?
      }
    mysql_close();
    ?>
<h1 align="center"> </h1>
  </body>
</html>
```

Et voilà ! Notre application est terminée.